

Computing Motion with 3D Memristive Grids

Chuan Kai Kenneth Lim^{1,*} and T.Prodromakis^{1,2,†}

¹*Centre for Bio-inspired Technology, Department of Electrical and Electronic Engineering,
Imperial College London, London SW7 2AZ, United Kingdom*

²*School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, United Kingdom*

(Dated: March 14, 2013)

Computing the relative motion of objects is an important navigation task that we routinely perform by relying on inherently unreliable biological cells in the retina. The non-linear and adaptive response of memristive devices make them excellent building blocks for realizing complex synaptic-like architectures that are common in the human retina. Here, we introduce a novel memristive thresholding scheme that facilitates the detection of moving edges. In addition, a double-layered 3-D memristive network is employed for modeling the motion computations that take place in both the Outer Plexiform Layer (OPL) and Inner Plexiform Layer (IPL) that enables the detection of on-center and off-center transient responses. Applying the transient detection results, it is shown that it is possible to generate an estimation of the speed and direction a moving object.

I. INTRODUCTION

Computing motion is essential for performing many daily tasks that also find applications in machine vision[1, 2], industrial automation [3–6] and robotics[7, 8]. Traffic engineers and transport authorities require motion computation and object tracking in order to implement effective intelligent transportation systems[9, 10]. For the defense industry, the ability to track mobile enemy targets[11] and control unmanned vehicles[12] is critical. The tracking of cell movements is also recently shown to play an important role in medical diagnostic procedures [13, 14]. While in the field of communications, combining motion estimation and tracking with predictive coding can allow for significant reduction in transmission bandwidth [15].

By utilizing parallel networks and hierarchical structures, biological systems are able to perform motion computation very efficiently[16]. In comparison, conventional motion computation systems typically utilize charged-coupled device (CCD) cameras and digital processors that result in time-sequential computations. As a result, large processing power is required to ensure that these serial computations can be performed in real time. In recent years, the need for physically smaller and more power efficient devices has driven research into neuromorphic[17, 18] and biomimetic systems[19], which are largely inspired by the parallelization naturally found in biological systems.

One of the first biomimetic motion computation systems was implemented by a team lead by Carver Mead in Caltech[20, 21]. By modeling the optical flow via a network of analog resistive devices, this approach was shown to be able to compute an estimate of the optical flow. Over the past two decades, various other biomimetic approaches to motion computation have been implemented

in CMOS technology[20, 22–34].

Perhaps, the most prominent implementation is a 128x128 asynchronous temporal contrast silicon retina developed by Delbruck[22]. This system makes use of address-event representation (AER)[35]; inspired by the way neurons utilise spiking events for communication. Essentially, the AER approach allows the transmission of only the local changes caused by movement in a scene, instead of transmitting at a fixed frame rate. This increases the computational efficiency of the chip, enabling it to perform over an impressive dynamic range of more than 120 dB with a low power consumption of 23mW.

Another notable implementation is Visio1[23], a retinomorphic chip with parallel pathways that mimics the OPL and IPL. Subthreshold current-mode circuits are used to model the autofeedback characteristics of horizontal cells (for spatial filtering) and the loop-gain modulation of amacrine cells (for adapting temporal filtering to motion). Detection of edges moving in one direction or the other is made possible by aVLSI implementation of ganglion cells that respond to motion in a quadrature sequence.

The recent discovery of practical memristive devices [36] has opened the option of improving on these CMOS motion sensors. It has been shown in [37] that memristor-MOS technology (MMOST) is capable of outperforming CMOS implementations in terms of power and size. Furthermore, the non-linear and adaptive response of memristors allow them to serve as excellent building blocks for the practical realisation of complex synaptic connections[38, 39]. Our group has previously demonstrated that memristive grids have a great potential for modeling certain aspects of early vision processing, which takes place in the OPL[40].

Here, we describe a novel approach, distinct from the current implementations [20, 22–34], that allows for motion computation based on memristive networks, by extending our results in [40]. A novel memristive thresholding scheme is utilized for enabling the tracking of moving objects within a scene. Also, a biomimetic model of both the OPL and IPL are described that facilitates the de-

* chuan.lim08@imperial.ac.uk

† t.prodromakis@soton.ac.uk

tection of ON and OFF center transient responses. In turn, we investigate the directional and speed computation capacity of our approach. Finally, we demonstrate that this platform can operate in a fault tolerant manner and can adapt to distinct lighting conditions, similarly to biological counterparts.

II. BACKGROUND

A. Vertebrate's Retina: Architecture and Cells

In order to perform any form of biomimicry, it is necessary to comprehend the architectural and functioning features of the leveraging biological system. This section serves as a brief unqualified introduction of the retina and is not meant to be a thorough review. More in depth information can be found in [41–43]. A simplified drawing showing the different layers of the human retina can be found in Fig 3 of [44] and Fig 1of [40].

There are 5 main types of retina cells (photoreceptor cells, bipolar cells, horizontal cells, amacrine cells and ganglion cells) laid along two major layers: the Outer Plexiform layer (OPL) and the Inner Plexiform layer (IPL). The photoreceptors (rods and cones) receive visual stimuli from the outside world. The outputs of the photoreceptor cells form synaptic connections with bipolar cells and horizontal cells. The horizontal cells are interneurons that form lateral local connections. The bipolar cells connect the OPL, comprising of horizontal and photoreceptor cells, to the IPL containing the amacrine cells and ganglion cells. Similar to horizontal cells, amacrine cells mediate synaptic lateral interactions between bipolar cells and ganglion cells in the IPL.

The photoreceptor cells essentially act as transducing elements, transforming the incident light stimuli into electrical signals. The magnitude of voltage change in the cells membrane is proportional to the logarithm of the intensity of light[29]. Horizontal cells form synaptic connections between photoreceptors and bipolar cells. Via the horizontal cells, lateral connections with neighbouring groups of photoreceptors and bipolar cells are made. These can be conceived as being a resistive layer spanning the OPL, with the architecture allowing dynamic range adjustments.

The output of a bipolar cell depends on the difference between the voltage at the photoreceptor it is connected to and the voltage at neighbouring photoreceptors, through the indirect connection via the horizontal cells. There are 2 kinds of bipolar cells. The ON-cells that respond to the onset of light and the OFF-cells that respond to the cessation of light. The photoreceptor, horizontal and bipolar cells do not generate action potential, instead, computation is performed via the flow of depolarization potentials. On the contrary, amacrine and ganglion cells in the IPL do generate spikes. Amacrine cells give transient light responses to either ON stimulus, OFF stimulus or both. Amacrine cells can be classified

into sustained and transient cells. Sustained amacrine cells obtain inputs from the bipolar cells that are not inhibited by other amacrine cells. The response of sustained amacrine cells then inhibits bipolar terminals in the narrow field region, causing those terminals to respond transiently. Essentially, this network of amacrine cells forms a high pass filter of the bipolar signal that facilitates the detection of moving edges[45].

Ganglion cells carry the output of the retina to the optical nerve, which leads to the visual cortex in the brain. Of the different types of ganglion cells, the directionally selective (DS) ganglion cells play a huge role in motion detection. There are three identified types of DS ganglion cells, namely ON/OFF DS ganglion cells, ON DS ganglion cells and OFF DS ganglion cells. ON/OFF DS ganglion cells perform the function of local motion detectors.

B. Mathematical Model of Memristors

The derivation of the mathematical memristor model employed here is largely guided by [46]. A memristive system was defined by Chua and Kang[47], following the original 1971 definition of the ideal memristor[48], as:

$$v = R(x)i \quad (1)$$

$$\frac{dx}{dt} = f(x, i) \quad (2)$$

This implies that the resistance $R(x)$ depends on an internal state x of the device while the time derivative of the internal state x is a function of x and i . The first physical model was conceived of a solid-state implementation by Strukov *et al.*[49], where x is proportional to charge q flowing through it. In the device, the magnitude of R can be modified reversibly between a highly conductive state R_{on} and a highly resistive state R_{off} , by modulating x :

$$R(x) = x(t)R_{on} + (1 - x(t))R_{off} \quad (3)$$

$x(t)$ is restricted to the interval $[0,1]$ and the time derivative of $x(t)$ is proportional to the current, as shown:

$$\frac{dx}{dt} = \frac{R_{on}}{\beta} i(t) \quad (4)$$

Substituting (3) and (4) into (1) we obtain:

$$v(t) = \beta \{x(t) + r[1 - x(t)]\} \frac{dx(t)}{dt} \quad (5)$$

where $r = R_{off}/R_{on}$ is the resistance ratio and β has a dimension of magnetic flux. Moreover, since $\varphi = \int v dt$, and by using $x \frac{dx}{dt} = \frac{1}{2} \frac{d}{dt} x^2$, it is possible to integrate both sides of (5) to obtain:

$$\varphi = \beta \left[-\frac{r-1}{2} x^2 + rx + c \right] \quad (6)$$

where c is a constant of integration determined by the initial conditions of x . This equation shows that flux is a quadratic function of the charge as x is proportional to q , hence, the non-linear relationship [36, 50].

Solid-state TiO_2 memristors implementations typically comprise two metal contacts that encompass a TiO_2 active core of thickness D . An external bias across the device causes charge to flow through the device. This causes a drifting of dopants resulting in the movement of the boundary between the two regions. By defining $w(t)$ as the coordinates of this boundary and μ_v as the average ion mobility, equation (4) becomes:

$$\frac{1}{D} \frac{dw(t)}{dt} = \mu_v \frac{R_{on}}{D^2} i(t) \quad (7)$$

and by integrating both sides, we obtain:

$$\frac{dw(t)}{D} = \mu_v \frac{R_{on}}{D^2} q(t) \quad (8)$$

It is observed that $x(t) = w(t)/D$ corresponds to the normalized width of the R_{on} region while $\beta = D^2/\mu_v$. Setting c to zero in equation (6) and substituting equation (8):

$$\varphi = -\frac{R_{on}\mu_v}{2D^2} \left(\frac{R_{on}}{R_{off}} - 1 \right) q^2 + R_{off}q \quad (9)$$

Assuming that $R_{off} \gg R_{on}$, the memristance of the device is obtained as:

$$M(q) = \frac{d\varphi}{dq} = R_{off} \left(1 - \frac{R_{on}\mu_v}{D^2} q \right) \quad (10)$$

Furthermore, as mentioned in [49], there are significant non-linearities in ionic transport, especially in the thin film edges at the boundary, which can be modeled by a window function, $f(x)$, on the right hand side of equation (7), giving:

$$\frac{dx(t)}{dt} = k i(t) f(x), k = \mu_v \frac{R_{on}}{D^2}, x(t) = \frac{w(t)}{D} \quad (11)$$

C. Optical Flow

The estimation of the direction and speed of a pixel in a sequence of image can be obtained from the optical flow; the approximation of local motion in an image by computing local spatial and temporal derivatives in a given sequence of frames. For a 2D picture, the optical flow indicates the speed and direction in which each pixel in an image moves between frames. The computation of optical flow is based on spatio-temporal intensity variations in brightness patterns. In this work we assume that all intensity variations are due to the motion of an object. In general, the optical flow and the velocity of an object are different. As explained in [51], a perfectly featureless rotating sphere has a non-zero velocity component, but it will not give rise to any optical flow. Whereas, a shadow

moving across the same sphere, which is now stationary, will produce an optical flow that is non-zero, although its velocity is zero. Nonetheless, apart from these situations, the computed optical flow is a good indication of the velocity of the moving object, provided a strong enough gradient between the moving object and its background exists. This is generally true for most natural scenes and it is assumed to be true for all input stimuli employed in this work.

1. 2D Motion Constrain Equation

Assuming $I(x, y, t)$ is the intensity of a brightness pattern at specific pixel located at coordinates (x, y) in the image at time t , and if this brightness pattern moves by $(\delta x, \delta y)$ within a timeframe δt , the intensity at the corresponding pixel will be $I(x + \delta x, y + \delta y, t + \delta t)$. Since this is essentially the same brightness pattern, it can be modeled as:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t) \quad (12)$$

And by employing a Taylor expansion about the point (x, y, t) , this reads:

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t + H.O.T \quad (13)$$

By comparing equation (12) and (13), it can be derived that $\frac{\partial I}{\partial x} \delta x + \frac{\partial I}{\partial y} \delta y + \frac{\partial I}{\partial t} \delta t = 0$ and $\frac{\partial I}{\partial x} \frac{\delta x}{\delta t} + \frac{\partial I}{\partial y} \frac{\delta y}{\delta t} + \frac{\partial I}{\partial t} \frac{\delta t}{\delta t} = 0$ and eventually,

$$\frac{\partial I}{\partial x} v + \frac{\partial I}{\partial y} u + \frac{\partial I}{\partial t} = 0 \quad (14)$$

This can be written more compactly as

$$(I_x, I_y) \cdot (v, u) = -I_t = \nabla I \cdot \vec{v} \quad (15)$$

∇I refers to the spatial intensity gradient and \vec{v} refers to the optical flow at pixel (x, y) at time t .

Equation (15) has 2 unknowns, which as explained in [51, 52] stem from the aperture problem. Insufficient information is available to measure the full image velocity, and only the component perpendicular to the edge or along the spatial gradient can be measured. In other words, a point in an image sequence only provides one independent image measurement whereas the velocity field has two components, forming an ill-posed problem.

2. Smoothness assumption

In order to solve this ill-posed problem, a second constrain is therefore required. Horn and Schunck [51] combined the 2D motion constrain together with a global

smoothness term, such the final velocity field \vec{v} is one which minimizes:

$$E(u, v) = \int \int (I_x + I_y + I_t)^2 + \lambda \left[\left(\frac{\partial u}{\partial x} \right)^2 + \left(\frac{\partial u}{\partial y} \right)^2 + \left(\frac{\partial v}{\partial x} \right)^2 + \left(\frac{\partial v}{\partial y} \right)^2 \right] dx dy \quad (16)$$

The first term implies that the solution should be as close as possible to the measured data while the second term imposes a smoothness constrain on the solution. The smoothness constrain derives from the fact that, apart from at specific discontinuities, adjacent points on an object have similar velocity and the brightness pattern varies smoothly almost everywhere. The magnitude of λ determines the importance of minimizing the smoothness term. With high Signal to Noise Ratio (SNR), the importance of the first term increases and λ will be small. In contrast, if the data is unreliable, λ will be larger, emphasizing the importance of the smoothness term.

Equation (16) shows that $E(u, v)$ is quadratic in the unknowns u and v . From standard variation calculus it can be shown that the corresponding Euler-Lagrange equations are linear in u and v :

$$\begin{aligned} I_x^2 u + I_x I_y v - \lambda \nabla^2 u + I_x I_t &= 0 \\ I_x I_y u + I_y^2 v - \lambda \nabla^2 v + I_y I_t &= 0 \end{aligned} \quad (17)$$

This results into two linear equations describing every point, which essentially capture the necessary components for calculating optical flow.

III. DETECTING MOVING EDGES WITH MEMRISTIVE GRIDS

We have previously presented (in Figure 4 of [40]) a retinomorph OPL, which modeled the synaptic interconnections between horizontal, bipolar and photoreceptor cells. Any light stimulus is translated into an appropriate voltage source to bias the underlying hexagonal memristive network via a series resistor. It was shown that the network was able to perform a local Gaussian filtering function similar to the biological counterpart [21], as well as edge detection via a memristor-based thresholding scheme. Nodes corresponding to edge pixels have large potential difference between neighbouring nodes due to the overlying intensity contrast at these pixels. From equation (10), the change in the memristance of the memristive fuse is proportional to the current flowing from that edge node to a neighbouring node. Therefore, by monitoring the state variance of memristive fuses associated with a node, edge detection can be performed. Here, we extend the results of [40] by performing detection of edges of moving objects from a video, investigating how a memristive network can be used to detect moving edges from a sequence of frames.

A. Simulation Methods

The employed video input stimulus is a 80x70 pixel, 0.6s segment of the video traffic.mju, which can be found in the Matlab Image Processing Toolbox[53]. At a frame rate of 15fps, this corresponds to 9 frames, shown as a montage in Figure 1a. Changes only occur at interval of the frame period(1/15s), and within this period the voltage stimulus at each pixel is a constant. This is modeled with a piecewise-linear voltage source at each node corresponding to a pixel.

Figure 1b describes how a single pixel in the video is converted into a voltage stimulus. In each gray scale frame, the intensity of each pixel is uniformly quantized into 256 levels, where 0 represents the absolute black and 255 represents white. Each pixel is therefore represented by an integer from 0-255. Since the maximum voltage input into each of the voltage input node in the memristive grid is set to 40mV, if the quantised intensity of a pixel is represented by X , the stimulus to its respective voltage input node is $X*40/255$ mV. This operation is performed for each of the pixel in each of the consecutive frames. The input voltage stimulus into a specific node corresponding to a pixel is then set to the piece-wise-linear combination of this quantised voltage level.

MATLAB was employed to implement the netlist for the hexagonal memristive network and to convert the input frames into equivalent biasing voltage stimulus. The generated netlist is then imported into HSPICE to perform the necessary circuit simulations on the constructed system. Memristive devices were simulated with Bi-olek's model [54], with the following parameter settings: $R_{on} = 100\Omega$ $R_{off} = 16k\Omega$ $R_{init} = 200\Omega$. In addition, the Prodromakis's windowing function [55] was employed to take into account the non-linear dopant kinetics of the memristive elements. The results from the HSPICE simulations were then imported back to MATLAB for further analysis.

B. Detection of moving edges via memristive state thresholding

The circuit was simulated for the input stimulus shown in Figure 1a. Edge detection was performed for each time step of the simulation, using the algorithm described in [40]. The results are shown in Figure 2b where the edges are marked by a white pixel. We notice that the memristance based thresholding scheme described in [40] was able to pick up the edges in the first 2 frames. However, for the rest of the frames, it seems like that edges were being picked up randomly. This is caused by the fact that the memristors used for the simulations are non-volatile and they do not reset or change before the next input frame is being simulated. Hence, even when the next frame is being simulated, the past memristance changes are still present, causing the results to be corrupted. In the following section, we address this issue by incorpo-

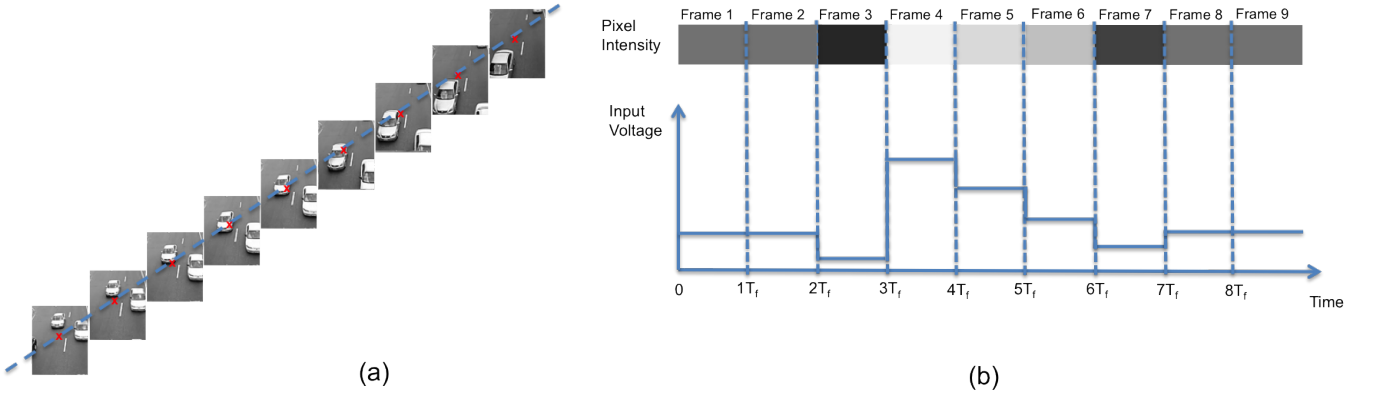


FIG. 1. (a) The sequence of the input frames is shown. A specific pixel in the video, marked by a Red X is indicated in each of the frames. The time between consecutive frames, T_f , is the inverse of the framerate. (b) The pixel intensity of each frame is shown and this corresponds to an input voltage stimulus, which is applied to the respective node in the memristive grid.

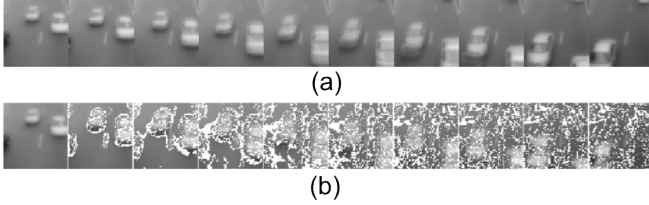


FIG. 2. (a) Evolution of the smoothed input stimulus. (b) Simulation of a sequence of images with the edge detection algorithm from [40]. Notice that edges were detected successfully only for the first 2 frames.

rating a novel memristive thresholding scheme that is applicable for detecting moving edges. A video showing the time evolution of the simulation is provided in S1 (see Supplementary Material[56]).

C. Edge detection based on monitoring the memristance modulation rate

To overcome the problem presented previously, we need to be able to separate the change in memristance corresponding to moving edges in the next frame from the changes in memristance corresponding to edges in the current frame. The proposed solution exploits the tracking of the temporal derivative of the memristance of the six memristive fuses connected to each node.

Following Equation (10), the change in memristance is proportional to the current flowing through the device, which is dependent on the potential difference across the device. Since each memristive fuse links adjacent nodes, the potential difference across it will be proportional to the intensity difference between the two nodes. If a node is an edge, there will be a strong intensity gradient with its neighbouring nodes. This results in a large

rate of change of memristance for the associated memristive fuses. Hence, by monitoring this rate of change, we are able to determine if a node corresponds to an edge pixel.

In a sequence of moving images, once the edge of the object moves away from a node, that particular node will cease to have a strong intensity gradient with its neighbouring nodes. Consequently, the temporal derivative of the memristance of the associated memristive nodes will no longer be high and hence the node will not be denoted as an edge. Figure 3 presents results that demonstrate our approach. A node is denoted as an edge pixel if the rate of memristance change of two or more memristive fuse associated with that node exceeds a predetermined threshold. Each edge pixel is denoted by a black pixel, superimposed on a white background. It is observed that the moving edges can now be successfully detected for the entire sequence of images. The full transient response of this experiment can be found in S2 (see Supplementary Material[56]).

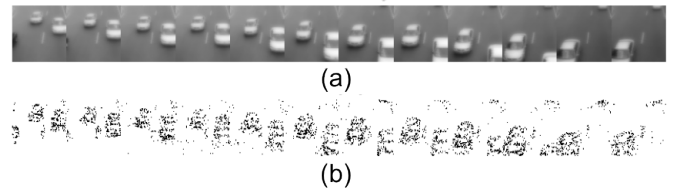


FIG. 3. (a) Evolution of the smoothed input stimulus. (b) Moving edge detection by tracking the temporal derivative of the memristance of the fuses associated to each node.

D. Adaptation to varying lighting conditions

Similar to [40], we evaluated the proposed algorithm against distinct luminance levels. Figure 4 demonstrates

that even when the brightness level was halved, the moving edges were still detected successfully. The moving edge detection results under the two different lighting conditions were compared pixel by pixel and the average mismatch between these two cases was found to be 4.22%. This is illustrated in Figure 4(c), where each white pixel corresponds to a mismatch. The full transient response of this experiment can be found in S3(see Supplementary Material[56]).

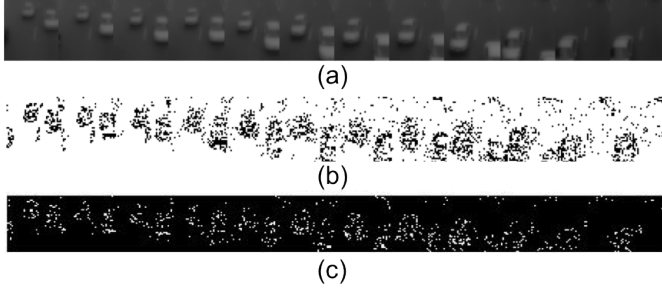


FIG. 4. (a) Evolution of the smoothed input stimulus. (b) Moving edge detection with input stimulus of half the original brightness. (c) Illustrates the accentuated mismatch between moving edge detection under normal lighting condition and under half the original brightness.

E. Memristance Variability Tolerance

The memristive network was tested to determine how tolerant it was to potential defects in the network. To simulate this, defect memristors were artificially introduced. Instead of simulating a circuit with all uniformly initialised memristors (i.e. $R_{on} = 100\Omega$ $R_{off} = 16k\Omega$ $R_{init} = 200\Omega$), we now initialise a percentage of the memristors randomly. These defected memristors will be initialised randomly with $R_{on} = 50\Omega - 100\Omega$ in increments of 10Ω , $R_{off} = 10k\Omega - 20k\Omega$ in increments of $0.5k\Omega$, R_{init} as a multiple (2-20) of the new R_{on} . The simulation results for 50% defects and 80% defects are respectively shown in Figure 5b and 5c.

The memristive network appears to be rather robust to faults, even when 80% of the devices were initialised randomly. The full transient response of this experiment can be found in S4(see Supplementary Material[56]). The results presented in Figure 5 demonstrates that the proposed system was still able to perform the detection of moving edges reliably. This is largely attributed to the fact that there are inherent redundancies in the network, as each node is interconnected to 6 neighbouring nodes. This is particularly important as the fabrication yield of memristive devices is relatively poor given that the memristors technological readiness is rather low. The scalability of memristive technology indeed opens the possibility of achieving reliability through redundancy.

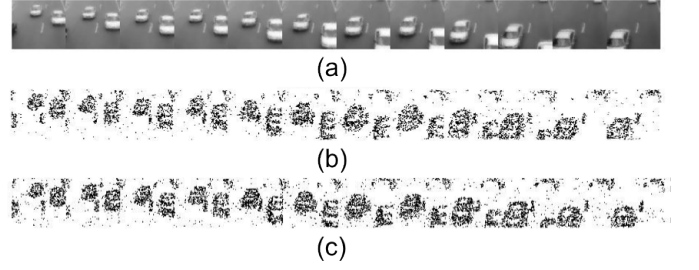


FIG. 5. (a) Evolution of the smoothed input stimulus. (b) Moving edge detection with artificially simulated faults of 50%. (c) Moving edge detection with artificially simulated faults of 80%.

IV. DOUBLE LAYERED MEMRISTIVE NETWORK

In the previous section, it was demonstrated that by employing the proposed thresholding scheme, memristive networks were able to perform moving edge detection, even with artificially simulated faults and at varying luminance levels. At the same time, this implementation is able to achieve local Gaussian filtering with a lower complexity as compared to other systems[40] and has been shown to perform better than resistive grids for edge preserving smoothing [57]. However, with this implementation, full optical flow computation is still not quite possible, as we have not yet obtained enough information from the sequence of moving images. The derivation of optical flow (in section II.C) shows that Equation (17) has to be solved in order to obtain the required optical flow. This requires both the spatial derivatives and the temporal derivatives at each pixel but our current implementation only provides us the spatial derivative (i.e. through the rate of change of memristance of memristive fuses). In this section, we present a 3D memristive grid that facilitates the computation of the temporal derivative.

A. Biomimetic OPL and IPL

From the results in the previous section, we note that a second memristive layer is required to model the amacrine cells within the IPL of the retina that are responsible for generating the transient responses. Our model does not explicitly emulate the functioning of the neurons in the IPL, but rather the synaptic interconnections between them. In order to generate the transient responses, a delay of the input stimulus is required, and this is introduced by the delayed version of the input in the second layer.

The 3D network, part of which is illustrated in Figure 6, consists of 2 hexagonal memristive layers interconnected with memristive fuses at each node. The current signal is applied to layer 1, which is responsible for computing the edges from the sequence of frames. Whereas,

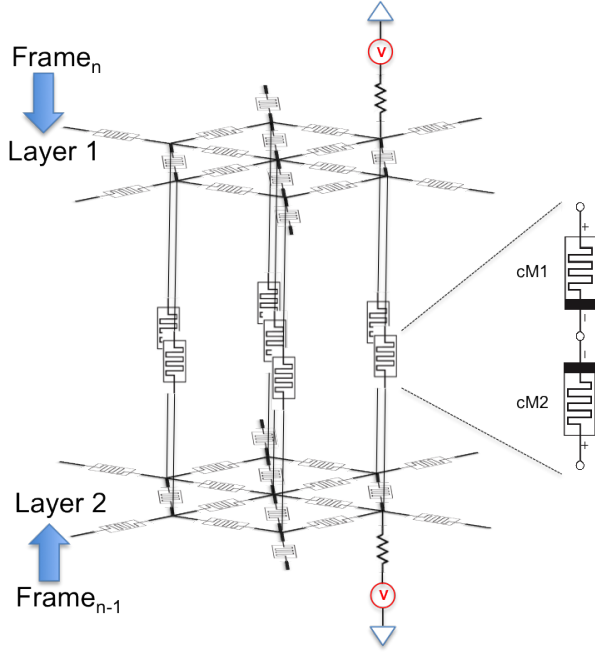


FIG. 6. Illustration of a section of the proposed 3D hexagonal memristive network model. Each node corresponding to a specific pixel is connected in a hexagonal manner to 6 neighbouring nodes via memristive fuses. The network consists of two interconnected layers. The current signal is applied to Layer 1 while a delayed signal of 1 frame is applied to layer 2. Each pixel in the input frame is converted into a piecewise-linear voltage value, which is then applied through a resistor to the memristive grid. For clarity, only two of these biasing connections are shown.

in layer 2, a delayed version of each frame is used, enabling the computation of the temporal derivative at each node via the memristance changes in the interconnecting memristive fuses.

Each connecting memristive fuse is made up of two individual memristors. We label the two memristor in each memristor fuse as Connecting Memristor 1 (cM1) and Connecting Memristor 2 (cM2), where cM1 is the memristor closer to layer 1 and cM2 is the one closer to layer 2.

When an object in the input stimulus is completely stationary, it implies that the pixels corresponding to that object remains the same for the current frame n and the delayed frame $(n-1)$. Consequently, the spatial voltage input (representing this object) to each layer will be identical and both ends of all connecting memristors are at equipotential. No current would thus flow and there will be no observable change in memristance.

Conversely, when an object in the input stimulus moves, the pixels corresponding to that object will move. This means that the current frame n is slightly different from the delayed frame $(n-1)$, reflecting this movement. Consider the case where a black box moves to the right by 2 pixels on a white background, as shown in Figure 7a. It can be observed that the rightmost and leftmost black

edges of the box will be moving by 2 pixels to the right. Comparing with frame $(n-1)$, the black box in frame n will be a shifted version by 2 pixels to the right. Hence, there will be a potential difference between the nodes corresponding to the new rightmost black edge (in frame n) and the old leftmost black edge (in frame $(n-1)$). At the new rightmost black edge, there will be a higher potential at layer 2 compared to layer 1. This causes a current flow through the connecting memristor fuse in from layer 2 to layer 1. Therefore, it is expected that the memristance $cM2$ will be increasing while the memristance of $cM1$ will be decreasing. Whereas for the old leftmost black edge, there will be a higher potential at layer 1 compared to layer 2. A current will flow from layer 1 to layer 2 causing the memristance of $cM1$ to increase and $cM2$ to decrease.

Therefore, by monitoring the memristance changes in each memristor of each connecting memristor fuse, it is possible to determine the movement of the edges. For a dark object moving on a light background, decreasing $cM1$ and increasing $cM2$ corresponds to an appearing edge while increasing $cM1$ and a decrease of $cM2$ corresponds to a disappearing edge. This is similar to the on-center response in the retina. On the other hand, for a light object moving on a dark background, increasing $cM1$ and decreasing $cM2$ corresponds to an appearing edge while decreasing $cM1$ and increasing $cM2$ corresponds to a disappearing edge. This is similar to the off-center response of the retina. Essentially, the response of these connecting memristors captures the response of the transient amacrine cells in the retina.

With a delayed version of the input into the second layer of memristive network, it is possible to derive the temporal derivative between consecutive frames. The temporal derivative between corresponding pixels in consecutive frames is inherently computed as changes in memristance of the memristors within the connecting memristive fuse. By combining the temporal derivative information with the spatial derivative information obtained from the hexagonal memristive network, it will be possible to separate moving objects from stationary objects. Essentially the hexagonal grid is performing computation of the sustained response while the connecting memristors are responsible for computing the transient response.

B. Emulating Transient Detection

1. Simple Inputs

As an initial proof of how the abovementioned motion computation can be done with the double-layered memristive network, two simple simulations were performed. Each of the simulation input stimulus consists of a 1s, 15 fps, 17x30 pixels video clip.

The first simulation, illustrated in Figure 7, involves a black box moving across a white background. From frame 1 to frame 5, the box is moving to the right at a

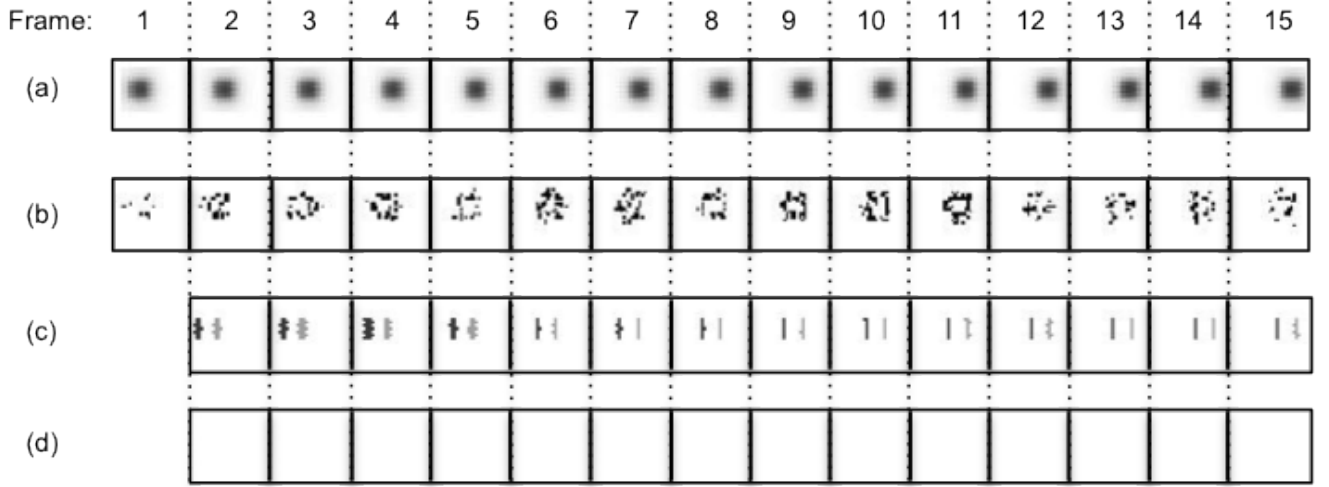


FIG. 7. (a) Evolution of input stimulus: A black box moving to the right at a rate of 2 pixels per frame from frame 1 to frame 5, and 1 pixels per frame from frame 6 to frame 15. (b) Simulation result of moving edge detection. (c) OFF-center transient detection results from monitoring the connecting memristive fuses. (d) ON-center transient detection results from monitoring the connecting memristive fuses.

rate of 2 pixels per frame and from frame 6 to frame 15, the box continues moving to the right at a slower rate of 1 pixel per frame. Since the moving object is dark compared to the surrounding, the on-center transient response (Figure 7(d)) is zero, which is expected. In the off-center transient response (Figure 7(c)), the light grey pixel corresponds to the appearing edge, while the dark pixel corresponds to disappearing edges. We see that the model is capable of monitoring the movement of the edges, while the speed of this movement can be derived from the thickness of the transient edges.

Our second example is similar to the first except for the fact that instead of a black box moving across a white background, a white box is now moving across a black background. In this case, the off-center transient response is zero and all the transient response is due to the on-center transient response. Here, the dark grey pixels correspond to an appearing edge while the light grey pixels correspond to a disappearing edge. The results are further illustrated in S5 (see Supplementary Material[56]).

2. Complex Inputs

The simulation was extended to a more complex input, which involves the first two frames of the traffic video stimulus shown in Figure 1a. Only two frames were simulated as the simulation of more frames are rather computationally intensive. The frame size used in this simulation was 80x70 pixel. This results in a circuit with 22400 nodes, 38602 memristive fuses, 11200 resistors and 11200 varying voltage sources. Even when running on a HP Z820 workstation with two 8-core Intel Xeon 2Ghz processor and 64 GB of RAM, the simulation environ-

ment (HSPICE) is incompetent in solving more extended cases.

The input stimulus, shown in Figure 8(a) consists of 2 frames. Frame 2 is overlaid on Frame 1, and for clarity, is shifted slightly to the right and it has purple colour saturation. From Frame 1 to Frame 2, it can be observed that the 2 white cars have moved slightly forward down the road, while the black car has almost moved off the image. The simulation was performed for 1 frame period of 1/15 s. By monitoring the memristance changes of the memristors within the connecting fuse, the results are shown on Figure 8(c) and (d) are obtained. From Figure 8(b), we do not consider the effects of the on-center or off-center response. The lighter grey dots correspond to decreasing cM1 and increasing cM2 while the darker grey dots corresponds to the increasing cM1 and decreasing cM2. This result shows the computation of transient response, where only moving objects were picked up.

In order to accurately compute the direction and speed of the object, it is necessary to separate the movement of dark edges from the movement of light edges. Essentially, the on-center response and the off-center response have to be distinguished. This can be done by splitting the result in acquired in Figure 8(b) based on the intensity of the input stimulus to each pixel. The off-center response is shown in Figure 8(c), where only the transient response due to a moving dark object/edge is shown. The darker dots correspond to a disappearing dark edge while the lighter dots correspond to an appearing dark edge. Similarly, the on-center response, showing the transient response due to a moving light object/ edge is shown in Figure 8(d). The darker dots now correspond to an appearing light edge while the lighter dots correspond to a disappearing light edge.

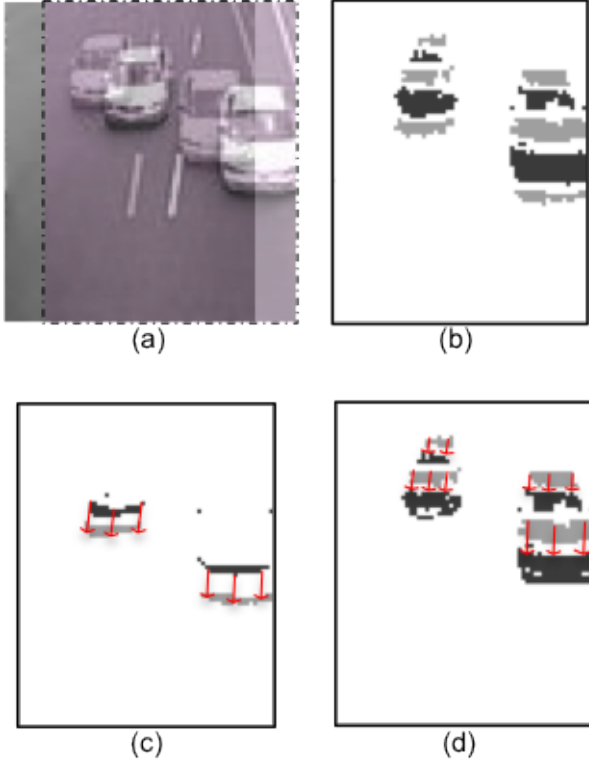


FIG. 8. Simulation of 2 frames from a video of traffic on a highway. (a) Input stimulus consists of 2 frames. Frame 2 is overlaid on Frame 1, and for clarity, it is shifted slightly to the right and it has purple colour saturation (b) Transient detection (c) OFF-Center response (d) ON-center response

C. Directional and Speed detection

Given the output from the transient detection, it is possible to use a 1-layer hexagonal memristive grid to compute the direction and speed of the moving object. The results from the transient detection shown in Figure 8(c) and (d) can be used as the input stimuli. To demonstrate this, the memristive grid was biased based on the off-center response from Figure 8(d). The darker pixels will correspond to a positive voltage source into the memristive network while the lighter pixels will correspond to a ground. Figure 9(a) illustrates the 80x70pixel hexagonal memristive grid where the voltage sources and ground are marked by “x” and “o” respectively. The rate of memristance changes of all interconnecting fuses are monitored and they are drawn in Figure 9(a) with a value dependent colour variation. Since it is expected that the current will take the shortest path from the source to the ground, the rate of memristance changes of the fuses will provide an indication of the direction and flow of the current. At each of the moving edge pixel, an estimate of the optical flow can be computed by the vectorial sum of the rate of memristance change of the 6 interconnecting fuses. Figure 9(b) and 9(c) respectively shows the results of this computation for Inset (a) and Inset (b) of Figure

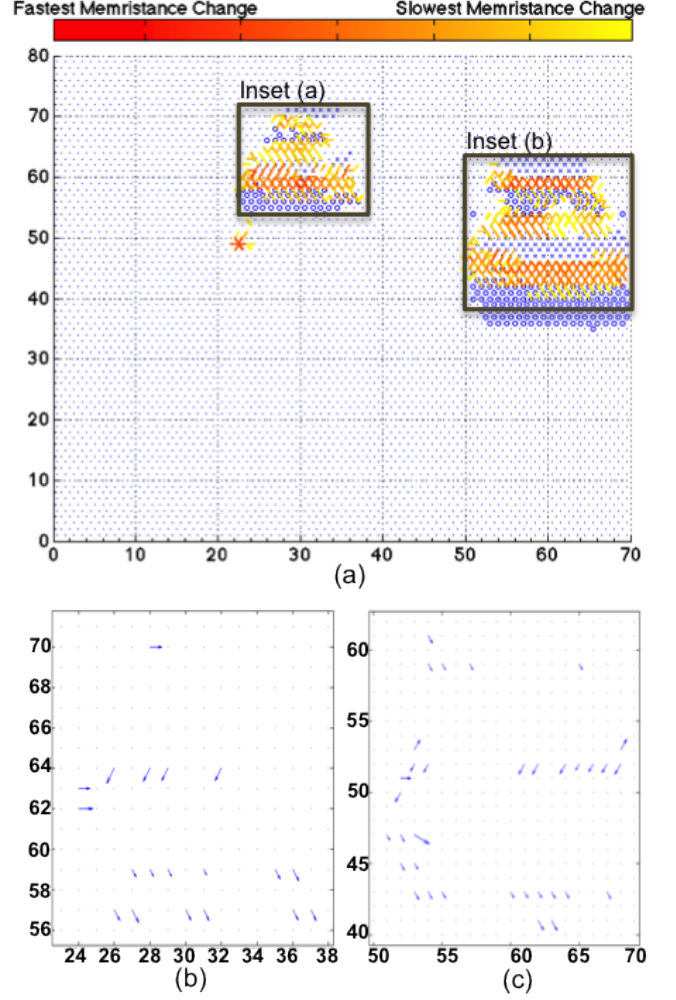


FIG. 9. (a) Results illustrating the rate of memristance change of fuses within the 80x70pixel network. Each memristive fuse is plotted with a color corresponding to the rate at which it’s memristance changes. (b) Illustrates the optical flow estimate from Inset (a). (c) Illustrates the optical flow estimate from inset (b).

9(a). Our method provides local optical flow estimates at the brightness edges, and as illustrated in Figure 9(b) and 9(c), was able to correctly determine that both cars were moving downwards.

V. CONCLUSION

In this paper, we presented a novel method for computing motion by leveraging on the scalability and non-linearity of memristors for emulating both the OPL and IPL of a vertebrates retina. The parallel processing nature of the proposed memristive network offers great potential in minimizing the computation and power requirements of conventional motion detection systems. A novel memristive thresholding scheme was introduced to allow

the detection of moving edges. In addition, a double-layered memristive network was constructed to enable to detection of ON-center and OFF-center transient responses. Applying the transient detection results into a 1-D memristive network, it was then shown that it is possible to generate an estimate of the speed and direction of the moving object. This study serves as a good proof of concept that memristive networks have indeed great

potential in performing motion computation effectively and efficiently.

VI. ACKNOWLEDGEMENT

The authors wish to acknowledge the financial support of the CHIST-ERA ERANet EPSRC EP/J00801X/1 and EP/K017829/1.

-
- [1] E. R. Davies, *Machine Vision: Theory, Algorithms, Practicalities* (Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, 2004).
 - [2] C. Steger, M. Ulrich, and C. Wiedemann, *Machine Vision Algorithms and Applications* (2008).
 - [3] E. N. Malamas, E. G. M. Petrakis, M. Zervakis, L. Petit, and J. didier Legat, *Image and Vision Computing* **21**, 171 (2003).
 - [4] L. Shafarenko, M. Petrou, and J. Kittler, *Trans.Img.Proc.* **6**, 1530 (1997).
 - [5] J. L. C. Sanz and D. Petkovic, *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **10**, 830 (1988).
 - [6] H. Li and J. C. Lin, *Industry Applications, IEEE Transactions on* **30**, 317 (1994).
 - [7] S. chii Liu and A. Usseglio-Viretta, in *in 2nd Int. ICSC Symp. Neural Comput* (2000) pp. 23–26.
 - [8] S. A. Huber, M. O. Franz, and H. H. Blthoff, *Robotics and Autonomous Systems* **29**, 227 (1999).
 - [9] M. A. Sotelo, F. J. Rodriguez, and L. Magdalena, *Intelligent Transportation Systems, IEEE Transactions on* **5**, 69 (2004).
 - [10] I. Masaki, *Intelligent Systems and their Applications, IEEE* **13**, 24 (1998).
 - [11] R. Canals, A. Roussel, J. L. Famechon, and S. Treuillet, *Industrial Electronics, IEEE Transactions on* **49**, 500 (2002).
 - [12] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, *Robotics and Automation, IEEE Transactions on* **19**, 371 (2003).
 - [13] D. W. Capson, R. A. Maludzinski, and I. A. Feuerstein, *Biomedical Engineering, IEEE Transactions on* **36**, 860 (1989).
 - [14] N. Ray, S. T. Acton, and K. Ley, *Medical Imaging, IEEE Transactions on* **21**, 1222 (2002).
 - [15] A. N. Netravali, in *Electrical and Electronics Engineers in Israel, 1989. The Sixteenth Conference of* (1989) pp. 0–18.
 - [16] S. Ullman, *Computer* **14**, 57 (1981).
 - [17] C. Mead, *Proceedings of the IEEE* **78**, 1629 (1990).
 - [18] Turel, J. H. Lee, X. Ma, and K. K. Likharev, *Int.J.Circuit Theory Appl.* **32**, 277 (2004).
 - [19] B. Bhushan, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **367**, 1445 (2009).
 - [20] J. Hutchinson, C. Koch, J. Luo, and C. Mead, *Computer* **21**, 52 (1988).
 - [21] C. Mead, *Analog VLSI and neural systems* (Addison-Wesley Longman Publishing Co., Inc, Boston, MA, USA, 1989).
 - [22] P. Lichtsteiner, C. Posch, and T. Delbruck, *Solid-State Circuits, IEEE Journal of* **43**, 566 (2008).
 - [23] K. Boahen and P. Bioengineering, in *INCREASING, ON, DECREASING, and OFF visual signals, Analog Integrated Circuits and Signal Processing* (Press, 2002) pp. 121–135.
 - [24] S. A. Baccus, B. P. Olveczky, M. Manu, and M. Meister, *The Journal of neuroscience : the official journal of the Society for Neuroscience* **28**, 6807 (2008), jID: 8102140; ppublish.
 - [25] R. G. Benson and T. Delbruck, in *Advances in Neural Information Processing Systems 4* (Morgan Kaufmann, 1991) pp. 756–763.
 - [26] A. Borst, *Current biology : CB* **21**, R990 (2011), cI: Copyright (c) 2011; JID: 9107782; CON: Curr Biol. 2011 Dec 20;21(24):2077-84. PMID: 22137471; ppublish.
 - [27] C.-F. Chiu and C.-Y. Wu, in *Circuits and Systems, 1997. ISCAS '97., Proceedings of 1997 IEEE International Symposium on*, Vol. 1 (1997) pp. 717–720 vol.1.
 - [28] T. Delbruck, *Neural Networks, IEEE Transactions on* **4**, 529 (1993).
 - [29] F. H. Eeckman, M. E. Colvin, and T. S. Axelrod, in *Neural Networks, 1989. IJCNN., International Joint Conference on* (1989) pp. 247–249 vol.2.
 - [30] S. Kameda and T. Yagi, in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, Vol. 4 (2003) pp. IV–792–IV–795 vol.4.
 - [31] S. Kameda and T. Yagi, in *SICE 2002. Proceedings of the 41st SICE Annual Conference*, Vol. 3 (2002) pp. 1853–1858 vol.3.
 - [32] R. C. Meitzler, K. Strohbehn, and A. G. Andreou, in *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, Vol. 3 (1995) pp. 2096–2099 vol.3.
 - [33] A. Mhani, G. Bouvier, and J. Herault, in *Solid-State Circuits Conference, 1995. ESSCIRC '95. Twenty-first European* (1995) pp. 326–329.
 - [34] W.-C. Yang, L.-J. Lin, H. Chiueh, and C.-Y. Wu, *Sensors Journal, IEEE* **11**, 3341 (2011).
 - [35] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision* (Kluwer Academic Publishers, Norwell, MA, USA, 1994).
 - [36] R. Williams, *Spectrum, IEEE* **45**, 28 (2008).
 - [37] I. E. Ebong and P. Mazumder, *Proceedings of the IEEE* **100**, 2050 (2012).
 - [38] B. Linares-Barranco, T. Serrano-Gotarredona, L. A. Camuas-Mesa, J. A. Perez-Carrasco, C. Zamarreo-Ramos, and T. Masquelier, *Frontiers in Neuroscience* **5** (2011).

- [39] T. Serrano-Gotarredona, T. Masquelier, T. Prodromakis, G. Indiveri, and B. Linares-Barranco, *Frontiers in Neuroscience* **7** (2013).
- [40] A. Gelencsér, T. Prodromakis, C. Toumazou, and T. Roska, *Phys. Rev. E* **85**, 041918 (2012).
- [41] B. A. Wandell, *Foundation of Vision*, 1st ed. (Sinauer Associates, Inc, United States of America, 1955).
- [42] J. E. Dowling, *The Retina - An Approachable Part of The Brain* (The Belknap press of Harvard university press, USA, 1987).
- [43] D. Regan, *Human Perception of Objects* (Sinauer Associates, Inc, Publishers, USA, 2000).
- [44] H. Kolb, *American Scientist* **91**, 28 (2003).
- [45] J. W. Lee, S. P. Chae, M. N. Kim, S. Y. Kim, and J. H. Cho, in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, Vol. 1 (2001) pp. 106–109 vol.1.
- [46] W. F.Y., *ArXiv e-prints* (2008), 0808.0286; Provided by the SAO/NASA Astrophysics Data System.
- [47] L. Chua and S. M. Kang, *Proceedings of the IEEE* **64**, 209 (1976).
- [48] L. Chua, *Circuit Theory, IEEE Transactions on* **18**, 507 (1971).
- [49] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, *Nature* **453**, 80 (2008).
- [50] I. Salaoru, T. Prodromakis, A. Khat, and C. Toumazou, *Applied Physics Letters* **102**, 013506 (2013).
- [51] B. K. P. Horn and B. G. Schunck, “Determining optical flow,” (1981).
- [52] J. L. Barron and N. A. Thacker, in *Tina Memo* (2005) Chap. 2004-012.
- [53] I. The MathWorks, *MATLAB Image Processing Toolbox - User’s Guide R2012b* (The MathWorks, Inc, 3 Apple Hill Drive, Natick, MA 01760-2098, 2012).
- [54] Z. Bielek, D. Bielek, and V. Biolkova, *Radioengineering* **18**, 210 (2009).
- [55] T. Prodromakis, B. P. Peh, C. Papavassiliou, and C. Toumazou, *Electron Devices, IEEE Transactions on* **58**, 3099 (2011).
- [56] “See supplemental material at <http://links.aps.org/supplemental/...> for videos illustrating the time evolution of simulations presented in this paper.”.
- [57] F. Jiang and B. E. Shi, in *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on* (2009) pp. 181–184.